

API.md

This document represents the API.md file which shows the keywords used in order to request or update data required by the represented application. This document also provides the URL required in order to connect with the remote server to perform query.

Retrieve user list

Retrieves the most updated current user list stored in the server. Any changes made in the game are later updated to this list to validate it.

- **URL**
/users
- **Method:**
GET
- **Data Params**
None
- **Success Response:**
Code: 200 OK
Content: [{"username": "dave", "password": "password_one"}, {"username": "john", "password": "password_two"}, {"password": "Aparant", "username": "AngryAppy"}]
- **Error Response:**
Code: 401 UNAUTHORIZED
Content: { error : "Log in" }
- **Sample Call:**

```
Sandbox.define('/users', 'GET', function(req, res) {  
  // retrieve users or, if there are none init, to empty array  
  state.users = state.users; // || [];  
  return res.json(state.users);  
});
```

Add a new user

Adds a new user to the current list of users. This is then updated in the game list as well as in the server

- **URL**
/users
- **Method:**
POST
- **Data Params**
[{"username": "dave", "password": "password_one"}]
- **Success Response:**
Code: 200 OK
Content: { status: "ok" }
- **Error Response:**
Code: 401 UNAUTHORIZED
Content: { error : "Log in" }
- **Sample Call:**

```
Sandbox.define('/users', 'POST', function(req, res) {  
  // retrieve users or, if there are none, init to empty array  
  state.users = state.users || [];  
  state.users.push(req.body);  
  return res.json({status: "ok"});  
});
```

Delete an existing user

Deletes an existing user from the current list of users. It is then updated in the game as well as on the server.

- **URL**
/users/{username}
- **Method:**
DELETE
- **URL Params**
Required:

username: [string]

- **Success Response:**

Code: 200 OK

Content: { status: "ok" }

- **Error Response:**

Code: 401 UNAUTHORIZED

Content: { error : {message: {username} } }

- **Sample Call:**

```
Sandbox.define('/users/{username}', 'DELETE', function(req, res) {  
  var user = _.find(state.users, { 'username': req.params.username });  
  if (!user) {  
    return res.json(404, { error: { message: req.params.username } });  
  }  
  // use Lodash reject to remove the user  
  state.users = _.reject(state.users, { 'username': req.params.username });  
  return res.json({status: 'ok'});  
});
```